DISCOVERING RESOURCES ON THE INTERNET

AUGMENTING FAST SEARCHES WITH METADATA

DC.Creator: Gisle Hannemyr
FAST.Version: 1.0ß2
DC.Date: 1999-05-17

FAST SEARCH & TRANSFER ASA BRYNSVEIEN 3B N-0667 OSLO NORWAY

Table Of Contents

1. Introduction

Fast Search and Transfer ASA (FAST) produces both hardware and software for rapid searches through huge quantities of data. One of the applications for the company's technology is Internet resource discovery.

FAST's approach to Internet resource discovery is initially similar to that of major brand name Internet search engines (e.g. AltaVista, Excite, HotBot, etc.). This approach can be summarised as follows:

1) A special program known as a «robot» (alternatively as a «scooter», «drone», «spider» or «crawler») explores the Internet and extract data about the resources it comes across;

2) the data extracted by the robot is stored in a data set on the search engine host computer (or computers) and is refined and structured to make it suitable for searches;

3) the data set is queried through some sort of user interface and the results of queries are presented as an ordered set.

FAST's search technology is already used to power several Internet search engines, including a service branded as "All the Web All the Time" (*http://www.alltheweb.com/*) and two specialized search services operated by Lycos: "FTP Search" (*http://ftpsearch.lycos. com/*) and "MP3 Search" (*http://mp3.lycos.com/*).

A recent study of search technology conducted by the author (Hannemyr 1999) identifies some problem specific to Internet resource discovery and proposes the use of metadata as a partial solution to some of those.

This technical report describes a framework for metadata to be implemented by FAST. It also discusses some practical issues in connection with the implementation, and gives some usage scenarios.

1.1 Target Audience

This primary audience for this technical note is FAST personnel. Its purpose is to serve as a reference and to provide guidelines for personnel responsible for implementing the metadata framework as part of search engines and related services.

It may also be used as a reference for resource owners and resource providers, who want to make use of the more precise search facilities and the improved display of search results that the use of metadata shall enable.

This technical note is not intended for end-users of the resulting services.

1.2 Acknowledgements

Thanks to Tor Egge and Knut Risvik for being very helpful in answering all my questions about the FAST FTP Search engine, and to Frode Lundgren and Haakon Wium Lie for introductions to XML and RDF, and to all the members of the DC Data Model Working Group (there are too many of you to list here) for answering a lot of silly questions about same. Thanks to John Lervik, Haavard Pettersen and Kimmo Tuominen for some very helpful comments on earlier drafts.

Any mistakes that remain are mine.

2. Basic Concepts and Requirements

Below some of the basic concepts and terminology used in this technical report is defined.

2.1 Internet Domains

Lynch (1997) has described the Internet as “a chaotic repository for the collective output of the world’s digital ‘printing presses’”. The Federal Networking Council (in a resolution passed on October 24, 1995) provides a more technical definition:

> *Internet* refers to the global information system that (i) is logically linked together by a globally unique address space based upon the Internet protocol (IP) or its subsequent extensions/follow-ons, (ii) is able to support communications using the transmission-control protocol/Internet protocol (TCP/IP) suite or its subsequent extensions/follow-ons and/or other IP-compatible protocols, and (iii) provides, uses or makes accessible either publicly or privatly, high-level services layered on the communications and related infrastructure described herein.

No reliable estimate of the number of host computers connected to the Internet exists, but the most recent estimate (January 1999) compiled from the Zone program reports maintained by Mark Lottor (Zakon 1999) puts the lower bound for host computers connected to the Internet around 43 million.

This report is concerned about using the Internet to discover A very comprehensive definition of a resource can be found in RFC2396, which describes the Uniform Resource Identifier (URI):

> A resource can be anything that has identity. Familiar examples include an electronic document, an image, a service (e.g., "today's weather report for Los Angeles"), and a collection of other resources. Not all resources are network “retrievable”; e.g., human beings, corporations, and bound books in a library can also be considered resources.
>
> The resource is the conceptual mapping to an entity or set of entities, not necessarily the entity which corresponds to that mapping at any particular instance in time. Thus, a resource can remain constant even when its content – the entities to which it currently corresponds – changes over time, provided that the conceptual mapping is not changed in the process. (Berners-Lee et al 1998)

A resource is an identifiable object of interest that can be identified by an URI.

To differentiate between the various types of resources accessible through the Internet, we introduce the following concept:

> An Internet *domain* is a collection of all the data resources publicly accessible through the Internet using a specific scheme or protocol (e.g. http, nntp, gopher, wais, ftp, z39.50). In addition to sharing a common protocol, the data within a particular domain usually also share other characteristics, such as formatting and presentation conventions, genre and access method.

Due to the phenomenal success of the World Wide Web in the recent years, there has been a pronounced tendency to equate the Internet with the World Wide Web. But technically speaking, the World Wide Web is just one of several domains on the Internet. Most major commercial search engines search the World Wide Web domain (based upon the http protocol), but there also exist other domains and services to search them. Examples include: Usenet (based upon the nntp protocol, and searchable through a service branded as Deja), gopher (searchable through Veronica, the Very Easy Rodent-Oriented Net-wide Index to Computerized Archives), WAIS (searchable through WAIS directory-of-servers) and ftp archives (searchable through a number of archie-servers and the FAST/Lycos FTP Search).

The two original services powered by FAST's Internet search technology (FTP Search and MP3 Search) search the ftp domain. The new FAST powered service branded "All the Web All the Time" searches the World Wide Web domain.

2.2 Data and Metadata and the Internet

Metadata is "data about data". Real life examples of metadata include such things as a library catalogue card (the "data" on the card describes the data contained in the books in the library) or a TV guide (the "data" in it describes the data in the programmes about to be broadcast.

Metadata is data that describes and qualifies other data. It is common to refer to the data described by metadata as a "resource". Typical examples of metadata are important properties of the resource (e.g. the name of the creator and the publisher, the year of publication), information required to locate the resource (e.g. the Dewey-code for a library book, and the time and channel for a television program), and data that is helpful when searching for the resource (e.g. a free-text description or a summary of the data, or a list of searchable subject keywords appropriate for the data).

A widely adopted model for metadata is called the Dublin Core Element Set. It is described in RFC2413 (Weibel et al 1998). The original set defines fifteen broad categories such as title, creator, publisher, date, etc. Work is currently under way in the Dublin Core community to provide means by which the fifteen elements of the Dublin Core may be expressed using HTML meta-tags (Kunze 1999), or using the Resource Description Framework (RDF) and encoded with the eXtensible Markup Language (XML) (Miller et al 1999).

The framework for FAST metadata described in this note is based upon the Dublin Core Element Set and the related work of expressing it in HTML and RDF/XML. The general idea has been to make as canonical use of this as possible, and to extend the model to solve problems particular to the FAST's requirements.

2.3 Binding Resources to Metadata

The metadata and the resource it describes may be separate entities (e.g. a library catalogue card and the book it describes) or the metadata may be embedded in the resource (e.g. the information that usually appears on the title page verso in a printed book). In the latter case the association between metadata and resource is implicit, but in the former case we must provide some sort of mechanism that manifests the association between the two entities. This mechanism is called binding.

In the present project, we shall use the following definitions:

Resource: An identifiable object of interest that is accessible and available to the public.

Metadata: A machine-understandable set of properties that aggregates to some description of a resource.

Binding: The mechanism that makes the association between metadata and the resource it describes manifest.

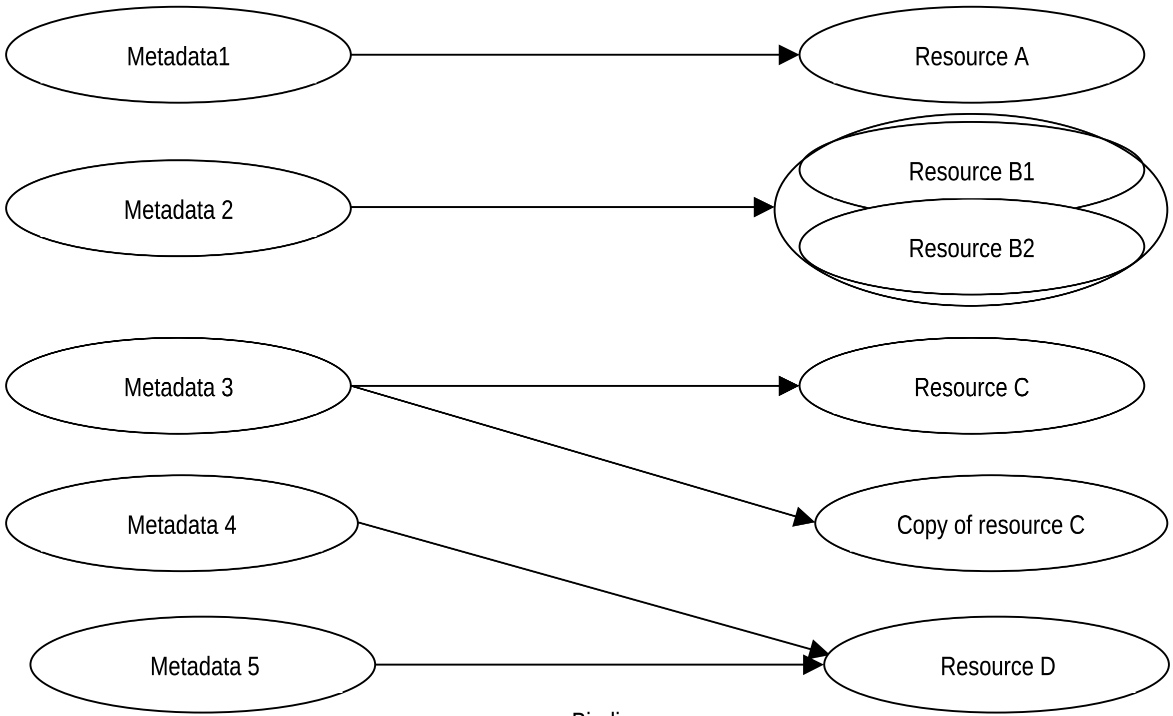
In the physical world of a library, the binding may be a Dewey-number that points the person browsing the library catalogue to the location on the bookshelves where the actual book described by the library card may be physically located. On the Internet, binding may be facilitated by means of a Universal Resource Locator (URL) that identifies the location of the resource on the Internet.

In practice, the resources The FAST project shall need to deal with will exist on the Internet. In principle, however, there should not be any problem in using the system to catalogue resources outside the Internet E.g. to create an Internet based Online Public Access Catalogue (OPAC) for a library service.

As noted, an Internet search engine harvests metadata from the net, and stores it in a local data set. The searchable data set is stored and maintained separate from the originating resource. Given the ephemeral nature of many Internet resources it is also a danger that a binding shall appear broken (if the resource is moved or disappear) or inaccurate (if the contents of the resource changes).

2.3.1 Cardinality

In a library, practical and physical constraints dictate that there is a single, directional relationship going from the catalogue card to the book, so there is a one-to-one relationship where a library catalogue card is bound a single instance of a book. This may also be the case with the most Internet web resources, but one can also imagine other binding arrangements. Four possible binding arrangements are shown in the diagram below, followed by a brief description:



Bindings

1) There is a one-to-one relationship between the metadata and the resource. I.e. one metadata set is bound to one, and only one resource.

2) A single metadata set is bound to a collection of closely related resources. This may be the case if we want to describe such as a whole file directory in the ftp domain, or an entire web site in the World Wide Web domain.

3) A single metadata set may be bound to multiple instances of a replicated resource. A common practice on the Internet is “mirroring”, whereby popular resources are copied to several alternate

locations. The reason for this redundancy is to provide multiple access points, and to provide backup access in case of network failures.

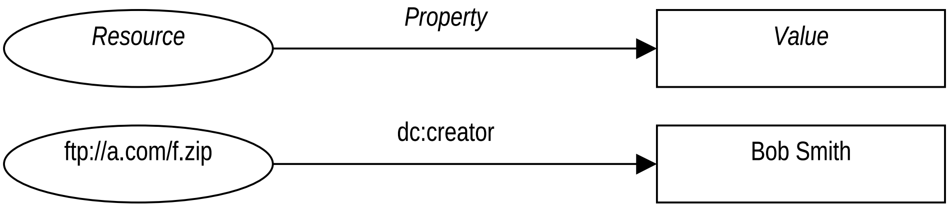
On the Internet, this sort of redundancy is a quite common occurrence, but it not "planned" or "designed" in the traditional sense. It simply arises through uncoordinated actions from different users and groups (Hannemyr 1999). Therefore, there is no point in having the means to express this particular relationship in the external scheme. Instead, we should have the means to *discover* replication, and an internal schema that lets us express what we have discovered.

4) Finally, we may also want to use two or more metadata sets to describe a resource. This is the situation when the resource may be viewed from two different perspectives (e.g. a Java applet that can be studied by a programming student to learn about some specific feature of the Java language, and *also* is a fun game that may be viewed as a nice gaming resource by someone completely uninterested in Java).

   Like replication, this is a common occurrence in the Internet. This particular relation may be a property of the resource deliberately designed by the creator of the resource, but it may also be due to some serendipitous qualities of the resource.

2.4 Basic Data Model

The model adopted for the FAST metadata framework is identical to the simple, triple-based model specified for RDF. The basic premise in this model is that an identifiable "*resource*" (e.g. a particular file or web page) may be described by a set of "*properties*" (e.g. creator, programming language, etc.), each of which has an associated *value* ("Bob Smith", "Java", etc). Below is the three elements of the model (*Resource, Property, Value*) is illustrated. Also shown is an instance, where the model is used to express the statement: "The resource named 'ftp://a.com/f.zip' is created by an entity named 'Bob Smith'".



The Dublin Core Initiative has taken upon itself to identify a core set of properties. The core element set is defined in RFC2413 (Weibel wt al 1998), and lists 15 basic elements or properties. This set consists of 15 elements is often referred to as DC 1.0, and has been stable since 1996.

2.5 Extending the Dublin Core

As indicated by the name, the 15 discrete properties of Dublin Core 1.0 are proposed as "core" properties. The fifteen elements fall short of providing all the metadata required by FAST. We therefore need to consider how to extend this core with application specific properties.

The authoritative description of DC 1.0 (Weibel 1998) does not indicate how the DC can be extended or adapted, but later developments, and RDF/XML (Brickley and Guha 1999) the recent mapping of DC into RDF/XML in particular (Miller et al 1999), provides some guidelines for this.

Below, I shall discuss two extension mechanisms: RDF Namespaces and DC Qualifiers.

2.5.1 XML Namespaces

RDF introduces a mechanism known as XML Namespaces (Bray et al 1999), that allows the definition of supplementary conceptual metadata frameworks by means of qualified names.

This simple mechanism allows alternative and supplementary metadata descriptions to coexist in the same domain without interfering with each other. Robots exploring a particular domain will be able to recognise metadata it is designed to process, and ignore any metadata put into that domain for other purposes.

Briefly: An XML Namespace is a mechanism for providing a human and/or machine-readable context for any resource description element. The context defines the semantics for the element, and qualifies the name so that it can be ignored by those not familiar with its context.

An XML namespace declaration can be done using the reserved XML attribute *xmlns:*. Following the attribute is a **namespace prefix** (which will be used to qualify the namespace in the rest of the document) and a **namespace name** (usually an URI). A string consisting of a namespace prefix and a name, separated by a colon, is referred to as a **qualified name.**

Below is an example of a sample metadata record for the resource *ftp://ftp.server.edu/bobbas12.zip,* created by *Bob Smith* using the *Java* programming language. The *creator* property is defined within the Dublin Core namespace (dc), and the *proglanguage* property is defined within the FAST (*fast*) namespace.

Please notice that entity names in XML are case sensitive. Lower case is usually used for qualified names.

```
<?xml version=”1.0”?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/TR/REC-rdf-syntax#"
  xmlns:dc="http://purl.org/dc/elements/1.0/"
  xmlns:fast=”http://www.ifi.uio.no/~gisle/fast/meta.html”>
  <Description xml:lang=”en” rdf:about=”ftp://ftp.server.edu/bobbas12.zip”>
    <dc:creator>Bob Smith</DC:Creator>
    <fast:proglanguage>Java</FAST:Proglanguage>
  </Description>
</RDF>
```

A similar concept is implied when using the HTML META tag to embed metadata in HTML (Kunze 1999). Here, the LINK tag is used to outline the name of schema (i.e. namespace prefix) and an URI of the defining document. Below is an example of how a resource may be described in HTML.

```
<LINK rel=”schema.DC” href=”http://purl.org/dc/elements/1.0/”>
<LINK rel=”schema.FAST” href=” http://www.ifi.uio.no/~gisle/fast/meta.html”>
<META NAME=”DC.Creator” CONTENT=”Bob Smith”>
<META NAME=”FAST.Proglanguage” CONTENT=”Java”>
```

HTML is not case sensitive, but Kunze (1999) still recommends that implementers follows his example “as regards prefix and element name capitalization”. This is the rationale behind using upper case “DC” and a capitalised “Creator” in the HTML example. Please also note that in XML, a colon is used to separate the prefix from the name, in HTML a full stop (usually pronounced “dot”) is used as separator. A property named “creator” within the “dc” namespace is written “dc:creator” in XML, and “DC.Creator” in HTML.

2.5.2 Qualified Dublin Core (DC 2.0)

Within parts of the Dublin Core community, there has been a strong desire to be able to qualify some of the elements. For instance, to qualify the role played by a *Creator* or *Contributor* (author, illustrator, photographer, etc.). In the 4th Dublin Core Workshop in Canberra (Weibel et al 1997) an extension for different types of qualifiers was introduced.

This extended version of Dublin Core is known as Qualified Dublin Core, or DC 2.0. The exact mechanisms for using DC with qualifiers are still debated in the Dublin Core community. A number of alternative proposals exist. An early version of DC inspired qualifiers materialised in the description of the HTML 4.0 META tag (Raggett et al 1998) and in some of examples listed in Kunze's (1999) draft RFC for HTML encoding of DC metadata. The more recent note from the Dublin Core data model working group (Miller et al 1999) proposes a mechanism for managing qualifiers based upon creating additional XML namespaces (e.g. *dcq* for DC qualifiers and *dct* for DC controlled terms).

One way of looking at qualifiers are through the terminology of object oriented design. We then have two types of qualifiers (element qualifiers and value qualifiers), that are used to express the two mechanisms known as inheritance and polymorphism:

1) Inheritance (element qualifier)

   Inheritance refers to the ability to create subclasses which "inherit" the properties from superclasses, but extend or refine those properties by specifying its own extension properties with added semantics (e.g. we may want to refine the meaning of the "date" element to mean "date accepted".

   Inheritance is experimentally expressed in HTML by means of a so-called "dot-notation", while in RDF, it is done by affixing the string "Type" to the DC element.

   Below is shown how to refine the date property to indicate acceptance date in HTML and in RDF:

```
<META NAME="DC.Date.Accepted" CONTENT="1999-05-10">

<dc:date>
 <rdf:Description>
  <rdf:value>1999-05-10</rdf:value>
  <dcq:dateType rdf:resource="http://purl.org/dc/terms/date/type/accepted"/>
 </rdf:Description>
</dc:date>
```

2) Polymorphism (value qualifiers)

   Polymorphism refers to the ability to override a property with one with the same signature (i.e. name) but with altered semantics. For example, the value of property "identifier* may be processed differently depending upon genre (e.g. *file, ISBN, URL* or *URI*). Other typical uses of polymorphism are to cater for different classification schemes (e.g. *ddc, LCSH* or *MESH*) used for the Subject property. In the FAST profile, we make use of polymorphism to permit alternate currencies as values for the Price property. In that case, one of the standard ISO 4217 three-letter abbreviations for the world's currencies (e.g. *USD, GBP, DEM, CHF,* etc.) is used.

   Polymorphism is catered for in HTML 4.0 (Raggett et al 1998) by means of the **scheme** attribute to the META tag and also by the following internationalisation qualifiers defined (ibid.) to support text in different languages:

– **Lang**: This qualifier specifies the language by means of an RFC1766 tag of the element value of the property field (i.e. not the language of resource itself).

– **Dir**: Direction of text (RTL or LTR).

Below is an example of use of a value qualifier in HTML and in RDF. In this example, we indicate that the date is encoded in a special profile of ISO 8601 with the distinguished name "WTN8601":

```
<META NAME="DC.Date SCHEME=”WTN8601” CONTENT="1999-05-10">

<dc:date>
 <rdf:Description>
  <rdf:value>1999-05-10</rdf:value>
  <dcq:dateScheme>WNT8601</dcq:dateScheme>
 </rdf:Description>
</dc:date>
```

Here is an additional example that uses the *Scheme* value qualifier to qualify a value in the FAST namespace used to express the price for a particular resource. The example indicates that the numeric amount for the price quoted is in Norwegian kroner:

```
<META NAME=”FAST.Price” SCHEME=”NOK” CONTENT=”499.50”>
```

Please note that the qualifier mechanisms still not fully resolved in the DC community. Reviewing a number of pilot DC applications that uses qualifiers reveal a number of incompatible and often confusing usages. Until the issue of qualifiers is better resolved, I would advise that very limited use be made of this feature.

2.6 Case Sensitivity

Unfortunately the treatment of case sensitivity and case insensitivity in XML and HTML is rather confused. As a matter of record, I've summarised the various rules I've come across below:

Case sensitive:

- XML entity names

- URIs (and most specifically the file name part of URLs).

- RFC1766 language codes. It is customary to give the language code in lower case, and the country code (if any) in upper case. Example:

```
<dc:title xml:lang="en-GB">What colour of money</dc:title>
<dc:title xml:lang="en-US">The color of money</dc:title>
```

Not case sensitive:

- HTML 4.0.

- Most XML values (i.e. generic identifiers, attribute names, IDs, IDREFs, name tokens in attribute values PI targets, notation names, and document type names).

- IANA registered distinguished named (e.g. character set identifiers, such as "ISO-8859-1")

Please note that there is no rule for case folding which works in the culturally expected manner for all speakers of all alphabetic languages: A lower-case e with acute accent is (correctly) uppercased one way in Quebec and a different way in metropolitan France. Lowercase I (with a dot) is uppercased one way in Turkish and another way in other languages using the Latin alphabet. Consequently, there are many in the XML community who feel that also XML values should be regarded as case sensitive. To avoid any problem with case folding, any case-insensitive entity must be expressed in ASCII (Unicode 0x00-0x7F) only.

2.7 Summary of Design Requirements

The FAST framework for metadata should make it simple and straightforward for authors, maintainers and users of Internet resources associate metadata with data on the Internet.

- The framework should general enough to be applicable to several alternate domains. Of particular interest is (http-based) web repositories and (ftp-based) file archives. But it should be possible to adapt the framework to other domains such as Usenet, Intranets, CD-ROMs and even proprietary resource libraries.

- When implementing the framework, the implementation should make use of state of the art technology (i.e. HTML-encoding of metadata and/or RDF/XML) in resource domains where this is appropriate. For ftp archives, we also need to provide a simpler and more straightforward format enabling authors to construct the metadata files themselves in a plain text editor without having any need for special authoring tools. This simple format should be both human and machine-readable. The alternate formats proposed for alternate domains should be based on the same framework.

- Metadata may be associated with a single resource (i.e. a single file, or a single web page), or a composite resource (a file ensamble, a directory subtree, or an entire web site). More than one metadata file may point to the same resource. Some Internet resources, in particular ftp archives, are frequently mirrored to other site, so we also have a situation where the same metadata set apply to “different” resources as far as an URI is concerned.

- Metadata should be easy to locate and identify for robots. There should be no way to confuse FAST metadata with ad hoc and legacy metadata or with metadata intended for some other robot.

3. The FAST Metadata Framework

The FAST Metadata Framework is based upon the Dublin Core (Weibel et al 1998, Dublin 1998), RDF/XML (Lassila and Swick 1999) and the proposed guideline for embedding metadata in HTML (Kunze 1999) and RDF/XML (Miller et al 1999).

3.1 Basic Element Set

The basic element set of the FAST metadata framework consists of the fifteen properties that make up the original Dublin Core Element Set as defined in RFC2413 (Weibel et al 1998) plus eight additional properties defined for the FAST namespace.

The fifteen members of the original (unqualified) Dublin Core Element Set are often referred to as DC 1.0. It has been stable since 1996. To this core element set, FAST has added eight properties: *version, proglanguage, sourcecode, price, md5, signature, environment* and *until.* The FAST properties are added to satisfy applications of FAST technology, and to support business logic that FAST foresees in future products.

In the table below, the fifteen elements taken from RFC2413 are shown shaded. They are listed here for with notes on recommended usage in the FAST environment. For details about these properties, including more specific semantics, qualifiers and common usage, please consult RFC2413 (Weibel et al 1998) and the evolving online profile of the Dublin Core (Dublin 1998). This FAST technical note should *not* be considered an authoritative source on the DC. RFC2413 and the Dublin Core profile are the authoritative references.

For the additional eight elements, FAST intend to create its own namespace with the prefix "FAST". This technical note (and its subsequent extensions/follow-ons) *is* the authoritative source for the semantics of properties defined in the FAST namespace.

All metadata elements, both in the DC namespace and in the FAST namespace, are optional and repeatable.

Namespace	Named properties	Description
DC	Identifier	A string or number used to uniquely identify the resource. When DC metadata is instantiated in HTML, the *identifier* property is used to bind the metadata to the resource. When DC metadata is instantiated in RDF/XML, a RDF property named *about* shall be used for this, making the *identifier* property redundant in RDF/XML entities that describe a single resource.
DC	Relation	An identifier of a second resource and its relationship to the present resource.
DC	Source	Information about a second resource from which the present resource is derived.
DC	Title	The name given to the resource, usually by the Creator or Publisher.
DC	Subject	Keywords describing subject. If several keywords are to be listed, the element should be repeated. In RDF/XML. In HTML, it is also acceptable to separate a list of elements using a semicolon.
DC	Description	Free text description of the content of the resource.
DC	Language	An RFC1766 tag identifying the language of the intellectual content of the resource.
DC	Coverage	The spatial or temporal characteristics of the intellectual content of the resource.
DC	Creator	Author (can be a person and/or an organisation) in plain text.
DC	Publisher	The entity responsible for making the resource available in its present form.
DC	Contributor	An entity not listed as Creator who has made intellectual contributions to the resource.
DC	Rights	A rights management statement or a link to such.
DC	Date	The date associated with the creation or availability of the resource. The default format to use is a profile of ISO8601 that is tentatively named WTN8601. This profile excludes two-digit years (unfortunately legal in ISO8601) and favours the format: YYYY-MM-DD. E.g.: the date 1999-05-17 corresponds to May 17, 1999.
DC	Type	The category of the resource (e.g.: software, font, rfc, image). Type should be selected from an enumerated list [to be specified].
DC	Format	The data format of the resource. (e.g.: PlainText, Applet, Application, TrueType, PDF, MP3), Format should be selected from an enumerated list [to be specified].
FAST	Version	Version number. It may be argued that the Dublin Core creator-supplied date field above makes this redundant. Version numbering is; however, such a fundamental aspect of software and other electronic resources that I feel that a version number should also should be included. Currently, a widespread practice on the Internet is to embed a version number in the URI. While this disambiguates the version number, it is search engine hostile given the fact that the metadata in the search engine may be out of sync with the actual resource. Having an explicit version number property allows for more permanent URIs even when frequent version superimposition takes place.

FAST	Proglanguage	Locating source code is one of the most important uses of a file-oriented search service. Therefore, this element needs to be included. For the sake of interoperability, proglanguage should be selected from a controlled list of programming languages.
FAST	Sourcecode	Is the source code included in the resource? (Legal values : YES, NO, PARTIAL)
FAST	Price	A monetary amount indicating cost of resource. Default is USD, but a qualifier listing a currency code according to ISO4217 may be used to indicate a different currency.
FAST	Md5	An MD5 digest of the resource file. This can be used by the search engine to discover replication (identical files mirrored at various sites) in order to provide the user with alternate sources in case of unavailability of one site. In addition, the publisher and/or creator can use the md5 field in a signed metadata file (see next element) to guarantee that the resource itself is authentic.
FAST	Signature	This field asserts that the signer guarantees the authenticity of the metadata document. This means that both the resource (providing it matches the md5 field) and the accompanying metadata is authentic. For instance, when looking for instances of the popular utility *pkzip*, one may specify that one is only interested in matches signed by Phil Katz (the publisher of *pkzip*). It uses the same netwide mechanisms for ensuring authenticity as DSig (Chu et al 1998).
FAST	Environment	The technical environment where the file is assumed to be suitable. This includes operating system, processor architecture, network setting (e.g. Novell, Internet, intranet, extranet), etc. For the sake of interoperability, environment should be selected from controlled list of environments. If we are looking for files to use on a Windows'95 system, we will want to filter out all other operating systems.
FAST	Until	A date when the resource should not be considered relevant any longer. The format is the same as the format for *date* property described above.

The functionality of some FAST properties can also be found in some of the metadata models now being developed, such as *PICS* (Miller 1996) and *indecs* metadata model for electronic commerce (Rust and Bide 1999).

The PICS framework cover such functionality as authentication, digital signatures and expiry dates, and the indecs project plan to tackle such issues as price, digital signatures, rights transactions and resource life cycle events. We have borrowed extensively from it, and plan to continue to do so, in particular when implementing an authentication framework.

The indecs framework is still at the draft stage, and as presented, it appears to be mostly unpopulated in these areas. The indecs framework also appears to be much more complex than what are needed at the present stage. Rather than adopting the very complex schema proposed by the indecs project, we have elected to define our extension properties using the simple, triple-based model specified for RDF (described in section 2.4 above). To maintain this basic simplicity has been an important design criteria.

3.2 Qualifiers

Given the unfinished state of qualified DC (aka. DC 2.0), we want to keep the use of qualifiers to a minimum. There are four qualifiers. The table below describes the qualifiers, and how they are expressed in HTML and in RDF/XML respectively:

HTML	RDF/XML	Description
charset	xml:encoding	An IANA registered name for the character set used in the value field. There can be one encoding for each document. The charset/encoding declaration would appear in the META tag returning UA data (http-equiv) declaration (HTML) or in the document type declaration appearing at the beginning of an external entity (RDF/XML)

lang	xml:lang	RFC1766 tag identifying the language used in the field it qualifies.
dir		Direction of text (LTR or RTL). Note that there is no need to supply a direction qualifier for RDF/ XML. XML uses the Unicode character set, where directionally of characters is defined as part of the standard.
dot-notation	dcq:*Type	Type qualifier.
scheme	dcq:*Scheme	Context the contents attribute should be interpreted in (e.g. *file*, *URL* or *URN,*). The scheme qualifier may also be used to indicate the currency for the *price* field. In that case, one of the ISO 4217 codes for the world's currencies (e.g. *USD, GBP, DEM, CHF,* etc.)

Below is an example of use of a qualifier in HTML. The example shows how to use the language qualifier to indicate that the *title* metadata field is in German:

```
<META http-equiv=”Content-Type” content=”text/plain; charset=ISO-8859-1”>
<META NAME="DC.Title" LANG="de" CONTENT="Römische Elegien">
```

Refer to section 4.1.1 for a more complete example of a metadata set embedded in HTML. For a similar example based upon RDF/XML, refer to section 4.3.1.

3.3 Binding in the FAST Metadata Framework

Referring back to the introductory discussion of cardinality (section 2.3.1) we recall that there are four different ways of defining the relationships between a set of metadata sets and a set of resources.

The first type (a one-to-one relationship where a single metadata set is bound to one and only one resource is the default, and requires no special considerations.

The second type describes a situation where a single set of metadata is bound to a collection of closely related resources (i.e. a composite resource). The composite may be such a thing as en entire website, a specific collection of discrete web pages (or “chapters”) that collectively make up an online “book”, or a collection of files that together make up a particular application in the ftp domain.

As all metadata properties (including *identifier*) are repeatable, our external schema supports resource collections by letting us explicitly list all members of the collection by means of repeating the identifier element. This (trivial) list is a syntactically valid external schema that expresses this one-to-many relation.

Such a scheme may, however, be less satisfying in practice. Explicitly listing all elements in a collection may be a lot of work, and it also introduces a very rigid relationship between the metadata and the resource (e.g. any restructuring of internal subentities of the composite resource must be reflected in an identical rear-rangement of the identifier properties in the metadata set).

Considering the practical implications of a one-to-many relation, I think it is useful to distinguish between those Internet domains that support some sort of hypertext, and those that do not. By means of hypertext, it is simple to create composite resources. Instead of binding to each and every element of the composite, we only need to bind to the “top” subentity, and let the internal structure (as expressed through hypertext links) of the collective delimit the composite. Internet domains that support hypertext include the World Wide Web (HREF), gopher (link files) and Usenet (References).

For domains without hypertext support, however, this approach is not possible.

The only important domain for FAST that does not support hypertext is the ftp domain (used by FTP search and MP3 Search). For this domain, FAST shall use its own metadata format (described below in section 4.2). A means to bind metadata to composite resources in the ftp domain will be described in that section.

The third type covers a situation where a single metadata set really describes multiple instances of a replicated resource. This situation typically arise due to the fairly common practice on the Internet of "mirroring" copies of popular resources are copied to several alternate locations. The reason for this redundancy is to provide multiple access points, and to provide backup access in case of network failures. As noted before, there is no need for expressing this relationship in the external schema. The internal schema, however, need to implement this. The FAST metadata model permits the search engine to deal with this eventuality implicitly, by means of the *Md5* property. This cardinality must be supported by the search engine's internal schema.

The fourth and final situation is where two or more metadata sets exist that describes a single resource. This situation does not arise from the external schema, but from the fact that anyone can take upon himself or herself to create a metadata set for a particular resource and place it on the Internet. The internal schema needs to be able to deal with this situation.

4. Three Metadata Formats

Below, three different metadata formats are described.

1) **HTML:** A format for expressing metadata using HTML meta-tags. It is suitable for resources resident in the World Wide Web domain.

2) **Simple adjunct text file:** A text file format suitable describing for non-Web resources. It is easy to create by hand.

3) **RDX/XML:** An experimental format based upon RDF/XML. It permits much a much richer data model than the two previous formats.

Which format of the three to use depends partly upon application, and partly upon resource domain.

4.1 HTML

Metadata may be embedded in web pages by use of the meta-tag as described in section 7.4.4 of the HTML 4.0 specification (Raggett et al 1998) and in (Kunze 1999).

As noted above (section 2.5.1) meta-tags must be qualified with a namespace prefix, which in HTML are defined using a special construct of the LINK tag. An HTML file with embedded meta tags is incomplete without at least *one* HTML LINK tag defining a profile (schema) for each different prefix. The two schemas we shall use for FAST metadata are "DC" (for the Dublin core), and "FAST" (for our eight additional properties). The LINK tags may look like this:

```
<LINK rel="schema.DC" href="http://purl.org/dc/elements/1.0/">
<LINK rel="schema.FAST" href=" http://www.ifi.uio.no/~gisle/fast/meta.html">
```

Using the *profile* attribute of the HTML HEAD tag, as proposed in the HTML 4.0 specification (Raggett et al 1998) is depreciated.

The LINK and META tags must appear inside the HEAD of an HTML file.

Each metadata element is optional and repeatable. Elements may appear in any order, and with no significance being attached to that order. Instead of repeating element, a semicolon may be used to list a number of items to be assigned a name. I.e. the fragment:

```
<META name="DC.Creator" content="John Lennon">
<META name="DC.Creator" content="Paul McCartney">
```

is semantically identical to:

```
<META name="DC.Creator" content="Paul McCartney; John Lennon">
```

Please also note the use of semicolons to separate items in a list. The use of commas for this purpose, as used in some examples in the HTML 4.0 specification (Raggett et al 1998), is depreciated.

4.1.1 Example

Here is an example of an HTML-file with a self-describing metadata set.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/transitional.dtd">
<html>
<head>
  <META http-equiv="Content-Type" content="text/plain; charset=ISO-8859-1">
  <LINK rel="schema.DC" href="http://purl.org/dc/elements/1.0/">
  <LINK rel="schema.FAST" href=" http://www.ifi.uio.no/~gisle/fast/meta.html">
  <META NAME="DC.Identifier" CONTENT="">
  <META NAME="DC.Title" LANG="de" CONTENT="Römische Elegien">
  <META NAME="DC.Title" LANG="en" CONTENT="Roman elegies">
  <META NAME="DC.Language" SCHEME="RFC1766" CONTENT="de">
</head>
<body>
   … body of Goethe poem …
</body>
</html>
```

4.1.2 Qualifiers

If necessary, qualifiers may be part of the meta element. Values are assigned to qualifiers by an equals («=» symbol), and should be quoted.

In this example the *lang* qualifier is used to specify the language for the value content attribute:

```
<META name="DC.Title" lang="fr" content="La Vie En Rose">
```

4.1.3 Binding

When expressing metadata in HTML, the DC.Identifier property is used for binding.

The following four examples shows four alternate ways of identifying embedded metadata in the self-describing web page with the fully qualified URI *http://www.mysite.net/home/mydoc.html*.

```
<META name="DC.Identifier" content="http://www.mysite.net/home/mydoc.html">
<META name="DC.Identifier" content="/home/mydoc.html">
<META name="DC.Identifier" content="mydoc.html">
<META name="DC.Identifier" content="">
```

The first three examples follow from the general rules for constructing URIs (Berners-Lee et al 1998). The final example shows the use of an empty identifier element to denote the fact that the web page is self-describing.

A web page may also describe an external resource, for example a file that resides in the ftp-domain:

```
<META NAME="DC.Identifier" CONTENT="ftp://ftp.ntnu.no/pub/rfc/rfc2413.txt">
```

And even a resource that is not available on the Internet:

```
<META NAME="DC.Identifier" SCHEME=”ISBN” CONTENT="1-56592-149-6">
```

4.2 Simple adjunct text file

The adjunct text file is created specifically for non-web resources (e.g. the Internet ftp domain that FAST currently makes searchable my means of the two services “FTP Search” and “MP3 Search”).

This data model for the simple adjunct text file is essentially the same as the data model for instantiation of metadata in HTML.

As noted in section 3.3, we need to make some special provisions to facilitate binding in a non-hypertext environment. This is discussed in section 4.2.3 below.

The syntax is simpler than the format used for embedding metadata in HTML. This is mainly done to facilitate creation of metadata by manual editing.

The metadata file is an ASCII text file. The file is simply a list of property/value tuples. The property is separated from the value by a colon, and there should only be one tuple per (logical) line. All whitespace is to be interpreted as a single space. All nametags must start in column 1. Whitespace in column 1 signify a continuation line (i.e. any physical line containing whitespace in column 1 is considered a logical continuation of the previous line).

The interpreter should be tolerant about what should constitute an end of line. Lines *should* be terminated with a CR LF pair (this is a line terminator that will create the least problems in viewers on all of Windows, Unix and Macintosh.

To screen out false positives, the FAST metadata file *must* start with a versioned «magic string». The only legal value at the time of writing is «%FAST File Search 1.0».

Default character set should be Unicode, UTF-8 but an optional MIME-style header similar to the one used in electronic mail may be inserted on line 2 an 3 to change this to something else (e.g. ISO-8859-1). The table below summarises how the two lines used to declare encoding should be used:

Named properties	Description
Content-Type:	Content type for metadata file (“text/plain” is the only content type currently supported). The Content-Type property must have an attribute named “charset” that defines the character set encoding used in the metadata file.
Content-Transfer-Encoding:	Encoding of metadata set (e.g. “QUOTED-PRINTABLE” or “8BIT”)

4.2.1 Example

Here is an example of a complete metadata file for a shareware game in the ftp domain:

```
%FAST File Search 1.0
Content-Type: text/plain; charset="iso-8859-1"
Content-Transfer-Encoding: 8BIT
DC.Title:        Bob's Baseball Spelling Game
DC.Creator:        Bob Frége
DC.Date:           1995-12-18
DC.Type:           Software
DC.Format:         Applet
DC.Identifier:     ftp://ftp.sample.org/pub/games/bobbas12.zip
DC.Language:       en
DC.Rights:         shareware
FAST.Price:        200; scheme="NOK"
DC.Subject:        Java; Educational; Games; Word Games
DC.Description:    Baseball animation helps you learn to spell.
  Great graphics. Requires sound support. This is a educational
  game for kids aged 6 to 9.
FAST.Proglanguage: Java
FAST.Sourcecode:   YES
FAST.Environment:  Internet
```

4.2.2 Qualifiers

The simple adjunct text file allows the same use of qualifiers as HTML. As in HTML qualifiers are written as `qualifyer="value"` (the value should be quoted). Qualifiers trail on the line, and are separated from the rest of the line by a semicolon.

As an example of use of a qualifier, refer to the example above, and the line containing the FAST.Price property. The scheme qualifier is here used to indicate that the price appearing as the value of the property is in Norwegian kroner, rather than the default US dollars.

4.2.3 Binding

In the ftp domain, binding is done by having the simple adjunct text file with the metadata, and the resource, resident in the same physical directory on the host computer. Binding is facilitated by means of simple naming conventions.

A metadata file for a singular file must have with the same filename as the singular file it describes, followed by the «magic suffix» *.mei* (MEta Information). Both files must reside in the same directory. A metadata file for a file named *foobar.zip* be named *foobar.zip.mei* and must be present in the same file directory as *foobar.zip.*

Composite resources, consisting of several files that we creator want to be located and treated as a composite, *must* be kept in a separate subdirectory tree. There shall be no files not belonging to the composite in any part of this subdirectory tree (i.e. one subdirectory tree shall correspond to exactly one composite resource). In this case, the file containing the metadata set, *must* have the «magic name» *fast_ftp.mei* (this filename as well as the "magic suffix" *.mei* case insensitive), The *fast_ftp.mei* file *must* reside in the root of the subdirectory tree of the composite resource.

It is recommended that a metadata set contain at least one DC.Identifier element.

This approach requires no software upgrade of ftp servers, but for it to succeed, it needs to be met with approval and compliance of authors of ftpable material. The authors must be willing to create meta information files according to the FAST metadata guidelines. There do, however, exist a strong incentive for the authors to do so. There is minimal work for them to create a meta information file, and doing so will greatly increase the chances of their file being found and downloaded by users genuinely interested in it.

4.3 RDS/XML

XML (eXtensible Markup Language) (Bray et al 1998) is has since 1996 been developed by the World Wide Web Consortium (W3C). Like HTML, XML is rooted in the SGML, the Standard Generalized Markup Language (ISO 1986). But while HTML only permits the use of a limited set of pre-defined tags, XML is extensible, and provides the means necessary to define new tags as required. This allows the definition of semantic tags that are very suited for computer-assisted resource discovery.

Rooted in XML, an experimental framework for resource description is being developed. This is known as the Resource Description Framework (RDF). RDF is introduced in (Miller 1998) as follows:

> The Resource Description Framework (RDF) is an infrastructure that enables the encoding, exchange and reuse of structured metadata. RDF is an application of XML that imposes needed structural constraints to provide unambiguous methods of expressing semantics. RDF additionally provides a means for publishing both human-readable and machine-processable vocabularies designed to encourage the reuse and extension of metadata semantics among disparate information communities. The structural constraints RDF imposes to support the consistent encoding and exchange of standardised metadata provides for the interchangeability of separate packages of metadata defined by different resource description communities.

RDF provides a very general framework for creating profiles that describes both the syntax and semantics of meta information. The RDF schema, model and syntax specification (Brickley and Swick 1999, Lassila and Swick 1999) is developed by W3C. The current status of this specification is *proposed recommendation*, so any implementation based upon RDF must be regarded as experimental.

An interesting application of RDF is known as PICS – Platform for Internet Content Selection (Resnick 1997). PICS is, among other things, a based upon distributed framework where authoritative metadata about a particular file or file ensemble are kept at a trusted host and referred to at the time of use. This appears to be fairly experimental at the moment, but it is of particular interest to us for two specific reasons: It is based upon a distributed model where anyone (in principle) can contribute metadata about specific resources. A system for authenticating metadata with digital signatures called DSig (Chu et al 1998) is being developed in conjunction with PICS.

While PICS is being developed as a means to block children’s access to «smut» on the World Wide Web, it can also be viewed as a general, application neutral method of binding authenticated metadata descriptions to distributed resources. There is, however, still a lot of work to be done before the RDF/PICS approach can be adopted as a basis for binding meta information to files.

Alternative XML-based approaches to metadata include MCF (Meta Content Framework) and XML-Data from Microsoft. These are also highly experimental and are not pursued at the moment.

4.3.1 Example

Below is an example of a tiny metadata set expressed by means of RDF/XML. In this example, we provide two alternate titles for a literary work, one in German, the other in English. In addition we make use of the DC language property to indicate that the resource itself (an online collection of poems by Goethe) is in German.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc="http://purl.org/dc/elements/1.0/"
    xmlns:dcq="http://purl.org/dc/qualifiers/1.0/">
  <rdf:Description rdf:about="http://www.doc.no/goethe/">
    <dc:title>
      <rdf:Alt>
        <rdf:li xml:lang="de">Römiche Elegien</rdf:li>
        <rdf:li xml:lang="en">Roman elegies</rdf:li>
      </rdf:Alt>
    </dc:title>
    <dc:language>
      <rdf:Description>
        <dcq:languageScheme>RFC1766</dcq:languageScheme>
        <rdf:value>de</rdf:value>
      </rdf:Description>
    </dc:language>
  </rdf:Description>
</rdf:RDF>
```

The above way of writing XML is know as *basic syntax*. It is fine for standalone XML entities, but not suited for embedding in HTML documents. For embedding RDF/XML in HTML documents, please refer to RDF abbreviated syntax, The abbreviated syntax is described in section 2.2.2 of Lassila and Swick (1999).

4.3.2 Qualifiers

Types and values are qualified in RDF/XML through use of the *dcq* namespace. Refer to section 2.5.2 for details.

4.3.3 Binding

When metadata is instantiated in RDF/XML, a RDF property named *rdf:about* is used to hold the value of an URI that binds the metadata set contained in the XML entity to the resource. As with HTML files, an empty value can be used in a self-describing web page.

Bibliography

Alvestrand, Harald: *Tags for the Identification of Languages;* RFC1766, March 1995, <http://ds.internic.-net/rfc/rfc1766.txt>

Berners-Lee, Tim and Robert Cailliau, Ari Luotonen, Henrik Frystyk Nielsen and Arthur Secret: *The World Wide Web;* Communications of the ACM, 37:8, August 1994, p. 76-82.

Berners-Lee, Tim and R. Fielding, U.C. Irvine, L. Masinter: *Uniform Resource Identifiers (URI): Generic Syntax;* RFC2396, August 1998, < http://www.ics.uci.edu/pub/ietf/uri/rfc2396.txt>

Bhushan, Abhay and Bob Braden, Will Crowther, Eric Harslem, John Heafner, Alex McKenzie, John Melvin, Bob Sundberg, Dick Watson, Jim White: *The File Transfer Protocol;* RFC172, MIT-Project MAC, 23 June 1971.

Bray, Tim, Dave Hollander and Andrew Layman: *Namespaces in XML,* W3C Recommendation January 14, 1999, <http://www.w3.org/TR/REC-xml-names/>.

Bray, Tim, Jean Paoli and C. M. Sperberg-McQueen: *Extensible Markup Language (XML) 1.0,* W3C Recommendation February 10, 1998, <http://www.w3.org/TR/1998/REC-xml-19980210>.

Brickley, Dan and R.V. Guha: Resource Description Framework (RDF) Schema Specification; W3C Proposed Recommendation, March 3, 1999, <http://www.w3.org/TR/PR-rdf-schema/>.

Chu, Yang-hua, Philip DesAutels; Brian LaMacchia, Peter Lipp: *PICS Signed Labels (DSig) 1.0 Specification;* W3C Recommendation, May 27, 1998, < http://www.w3.org/TR/REC-DSig-label/>

Dublin: *Description of Dublin Core Elements*, undated, viewed oct. 30, 1998, undated <http://purl.oclc.-org/metadata/dublin_core_elements>

Emtage, Alan and Peter Deutsch: *Archie—an Electronic Directory Service for the Internet,* USENIX Winter 1992 Technical Conference Proceedings. San Francisco: USENIX Association, 1992. 93-110.

FAST: *Information Memorandum,* May 15, 1998

Hakala, Juha et al: *The Nordic metadata project. Final Report,* Helsinki University Library, June 1998, <http://linnea.helsinki.fi/meta/nmfinal.htm>

Hamilton, Martin and Jon Knight: *Metadata & Cataloguing on the World Wide Web;* Invited presentation at the third INSIGHT workshop, November 1996, <http://www.roads.lut.ac.uk/Presentations/IN-SIGHT-Cataloguing/cataloguing.html>

Hannemyr, Gisle: *Conceptualizing Resource Discovery on the Internet;* submitted to the Hawaii International Conference on System Sciences (HICSS33), 1999.

Hauben, Michael and Ronda Hauben: *Netizens. On the History and Impact of Usenet and the Internet;* IEEE Computer Society Press, 1997

ISO: *ISO 8879:1986(E). Information processing – Text and Office Systems – Standard Generalized Mark-up Language (SGML);* First edition – 1986-10-15, Geneva; International Orgaization for Standardiza-tion, 1986.

Kunze, John: *Encoding Dublin Core Metadata in HTML,* Internet-Draft (Informational), March 18, 1999.

Lassila, Ora and Ralph R. Swick: *Resource Description Framework (RDF) Model and Syntax Specificat-ion,* W3C Recommendation, February 22, 1999, < http://www.w3.org/TR/REC-rdf-syntax/>

Lawrence, Steve and C. Lee Giles: *Searching the World Wide Web;* Science, Vol. 280, 3. april 1998, p. 98-100

Lynch, Clifford: *Searching the Internet;* Scientific American, p. 44-48, March 1997

Miller, Jim (ed.): *PICS Label Distribution Label Syntax and Communication Protocols;* W3C Recommen-dation, 31-October 1996, <http://www.w3.org/TR/REC-PICS-labels>

Miller, Eric: *An Introduction to the Resource Description Framework;* D-Lib Magazine, May 1998

Miller, Eric, Paul Miller and Dan Brickley: *Guidance on expressing the Dublin Core within the Resource Description Framework;* 1999-05-13, <http://www.ukoln.ac.uk/metadata/resources/dc/datamodel/-WD-dc-rdf/>

Raggett, Dave, Arnaud Le Hors and Ian Jacobs: *HTML 4.0 Specification;* W3C Recommendation, April 24, 1998, <http://www.w3.org/TR/REC-html40/>

Resnick, Paul: *Filtering Information on the Internet;* Scientific American, March 1997

Rust, Godfrey and Mark Bide: *Indecs metadata model;* Draft, April 7 1999, <http://www.indecs.org/pdf/-model1.pdf>

Weibel, Stuart, Renato Ianella and Warwick Cathro: *The 4th Dublin Core Metadata Workshop Report;* D-Lib Magazine, June 1997, <http://www.dlib.org/dlib/june97/metadata/06weibel.html>

Weibel S., J. Kunze, C. Lagoste, M. Wolf: *Dublin Core Metadata for Resource Discovery;* RFC2413, September 1998, <ftp://ftp.ntnu.no/pub/rfc/rfc2413.txt>

Zakon, Robert Hobbes: *Hobbes Internet Timeline v4.1;* undated,viewed 1999-04-18, <http://info.isoc.org/-guest/zakon/Internet/History/HIT.html>